

SEHH2239 Data Structures

Mid-term Revision

Question 1

Figure 9 shows the linked list structure in which the pointer head points at the first node and the pointer next points at the next node.

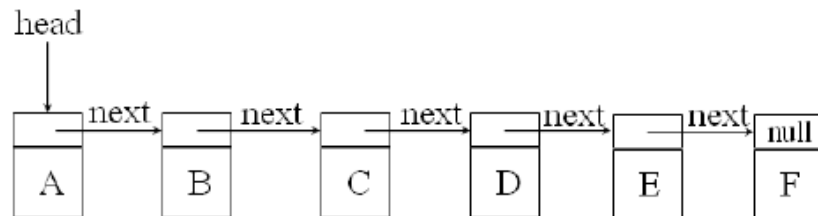


Figure 9

(a) After executing the following Java statements,

```
head.next.next.next.next.next.next = head.next.next;
head.next.next = head.next.next.next;
head.next.next.next.next.next.next = head;
head = null;
```

the linked list in Figure 9 will change to a circular list. Draw that circular list. (3 marks)

(b) Complete the following Java statements such that after executing these statements, the linked list in Figure 9 will change to the two circular lists in Figure 10 and the pointer head will become a null pointer. (Note: modifying any of the following given parts of the Java statements or adding Java statements is not allowed.) (4 marks)

H. n. n. n. n. n
H. n. n. n. n. n
head.next.next.next.next.next.next = head.next.next.next.next;
head.next.next.next.next = head.next.next.next;
head.next.next.next = head.next.next.next;
head = null;

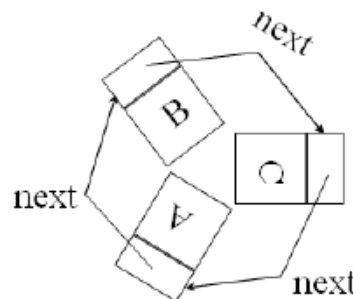
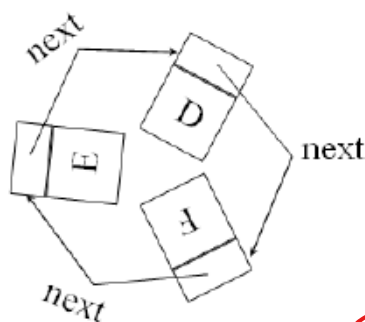
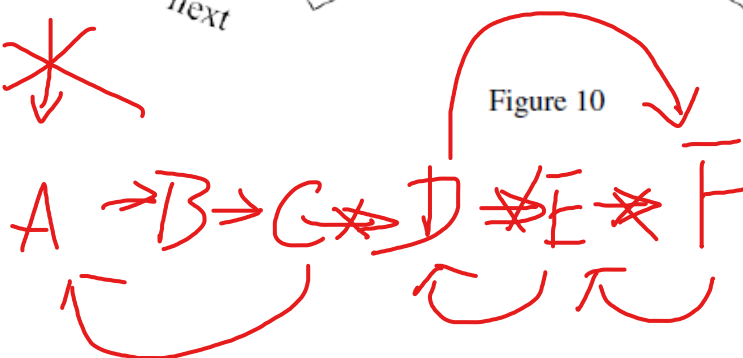


Figure 10



Question 2

- (a) Which sorting algorithm is easily adaptable to singly linked lists? Explain your answer. (5 marks)
- (b) Sort the following sequence of keys using merge sort and write down clearly the sequences in each pass. (5 marks)

66, 77, 11, 88, 99, 22, 33, 44, 55

Question 3

- (a) Given the following sort trace, identity which type of the probably sorting algorithm it is using with and explain your answers.

(i) **Sorting Algorithm 1**

(3 marks)

[13, 24, 14, 16, 60, 39, 8, 17, 50, 10]
[13, 14, 24, 16, 60, 39, 8, 17, 50, 10]
[13, 14, 16, 24, 60, 39, 8, 17, 50, 10]
[13, 14, 16, 24, 60, 39, 8, 17, 50, 10]
[13, 14, 16, 24, 39, 60, 8, 17, 50, 10]
[8, 13, 14, 16, 24, 39, 60, 17, 50, 10]
[8, 13, 14, 16, 17, 24, 39, 60, 50, 10]
[8, 13, 14, 16, 17, 24, 39, 50, 60, 10]
[8, 10, 13, 14, 16, 17, 24, 39, 50, 60]

(ii) **Sorting Algorithm 2**

(3 marks)

[30, 12, 64, 21, 46, 29, 21, 46, 68, 17, 11, 36, 9]
[30, 12, 64, 21, 46, 29, 21, 46, 68, 17, 36, 11, 9]
[30, 36, 64, 21, 46, 29, 21, 46, 68, 17, 12, 11, 9]
[30, 36, 64, 21, 46, 29, 21, 46, 68, 17, 12, 11, 9]
[30, 36, 64, 21, 46, 29, 68, 46, 21, 17, 12, 11, 9]
[30, 36, 64, 46, 46, 29, 68, 21, 21, 17, 12, 11, 9]
[30, 36, 64, 46, 46, 68, 29, 21, 21, 17, 12, 11, 9]
[68, 36, 64, 46, 46, 30, 29, 21, 21, 17, 12, 11, 9]
[68, 46, 64, 46, 36, 30, 29, 21, 21, 17, 12, 11, 9]
[68, 46, 64, 46, 36, 30, 29, 21, 21, 17, 12, 11, 9]
[68, 64, 46, 46, 36, 30, 29, 21, 21, 17, 12, 11, 9]
[68, 64, 46, 46, 36, 30, 29, 21, 21, 17, 12, 11, 9]

Question 4

Question C1

Given the following Python code:

```
import random

list1 = [20, 36, 16, 36, 7, 48, 67]

def display():
    print(list1)

def randomPosIntArr(n, max):
    if n < 0 or max < 2:
        print("n cannot less than 0 or max cannot less than 2")
    for i in range(0, n):
        list1[i] = random.randint(0, max)
    print(list1[1:7])

# code to be completed

print("Original List:")
display()

print("Reverse List:")
reverse()

print("Truncated List with n = 3:")
truncate(3)

print("Random List with 6 elements and max is 60:")
randomPosIntArr(6, 60)

print("Random List with 10 elements and max is 15 with no
duplicate:")
uniqueRandomInts(10, 15)
```

(a) Linear list can be implemented by a list in Python. Add the following methods to perform the required functions for the list:

(i) `truncate(n)` that print the shortened list by keeping only the first `n` elements of `list1`.

(4 marks)

(ii) `reverse()` that print the reverses the order of the element in `list1`.

(4 marks)

(iii) `uniqueRandomInts(n, max)` that print a new list named `list2` to become an array of length `n` with **unique** nonnegative random integers that are less than `max`.

(8 marks)

(b) Show the output when successfully executing the given Python code.

(4 marks)

Note you may show only the added Python code in your answer.